Поиск и замена с помощью регулярных выражений в notepad++

Содержание:

[Метасимволы и их значение]
[Примеры применения метасимволов для поиска]
[Примеры применения регулярных выражений для поиска и замены]
[Как очистить историю «поиска» в Notepad++]

Регулярное выражение – это шаблон текста, который состоит из обычных символов (например, букв от а до я) и специальных символов, известных как метасимволы. Шаблон задает правила поиска и может осуществлять различные манипуляции с текстом.

Далее приведены метасимволы и их значение, а затем уже примеры применения.

Метасимволы и их значение

Метасимволы и их значение

Символ	Описание
.	Точка представляет один любой символ
^	Начало строки
$	Конец строки
^$	Пустая строка (начало и конец, между которыми пусто)
.+	Любая не пустая строка
\s	Пробел
\S	Не Пробел
\w	Буква, цифра или подчеркивание
\d	Любая цифра
\D	Любой символ, но не цифра
[0-9]	Любая цифра
[a-z]	Любая буква от a до z (латинский набор символов) в нижнем регистре [1]
[а-я]	Любая буква от а до я (русский набор символов) в нижнем регистре [1]
[A-Z]	Любая буква от a до z в ВЕРХНЕМ регистре [1]
[А-Я]	Любая буква от А до Я в ВЕРХНЕМ регистре [1]
*	«Повторитель». Означает, что предшествующий символ может повторяться (0 или более раз)
.*	Абсолютно любой набор символов. Например, условие .* найдет любой текст между тегами
(^.*$)	Любой текст между началом и концом строки

Примеры применения метасимволов для поиска

Примеры применения метасимволов для поиска

Пример	Описание
([0-9][0-9]*.)	ищет любые цифры, в данном случае двухзначные цифры
\n	ищет символ новой строки
\r	ищет пустые строки содержащий символы «перевод каретки»
^$	ищет пустые строки
\n\r	ищет пустые строки содержащий символы – символ новой строки и «перевод каретки»
\s	ищет класс пробельных символов. К пробельным символам относятся пробел, символ табуляции, возврат каретки, символ новой строки и символ перевода страницы. То же самое, что и [\t,\r,\n,\f]
\S	ищет класс не пробельных символов. То же самое, что и [^ \t,\r,\n,\f]
^\s*$	ищет пустые строки содержащие пробел [2]

^[ ]*$	ищет пустые строки содержащие пробел [2]
^Тест	ищет слово Тест в начале строки
Тест$	ищет слово Тест в конце строки
\bтом	ищет набор символов том только в начале слов, то есть в слове Томск будет найдено, а в слове Потом нет
том\b	ищет набор символов том только в конце слов, то есть в слове Томск не будет найдено, а в словеПотом будет
\Bтом\B	ищет набор символов том не в начале и не в конце слов, то есть в слове Томск не будет найдено, и слово Потом также будет проигнорировано, а вот в словах Автоматизация, Хрестоматия будет найдено
Маша|Оля	ищет все заданные слова в тексте. Например, нужно найти определенные имена в тесте - *Маша и Оля, вечером собирались в кино, но встретив Сашу и Пашу, Маша ушла с Пашей в концерт, а Оля с Сашей решили погулять в парке*. Найдет все слова Маша и Оля, то есть четыре совпадения.

[1, 2] Часть примеров уже устарела, notepad++ совершенствуется и часть примеров делается стандартными командами. Например, конструкции вида **[а-яА-Я]** - *искать в любом из регистров* потеряли смысл, а явное указание на регистр при снятом чек-боксе будет проигнорировано и будут найдены все вхождения в верхнем и нижнем регистрах.

«Удалить пустые строки содержащие пробел» сейчас выполняется через меню «Правка» – «Операции со строками». Но с их помощью можно разрабатывать свои шаблоны поиска и замены.

Примеры применения регулярных выражений для поиска и замены

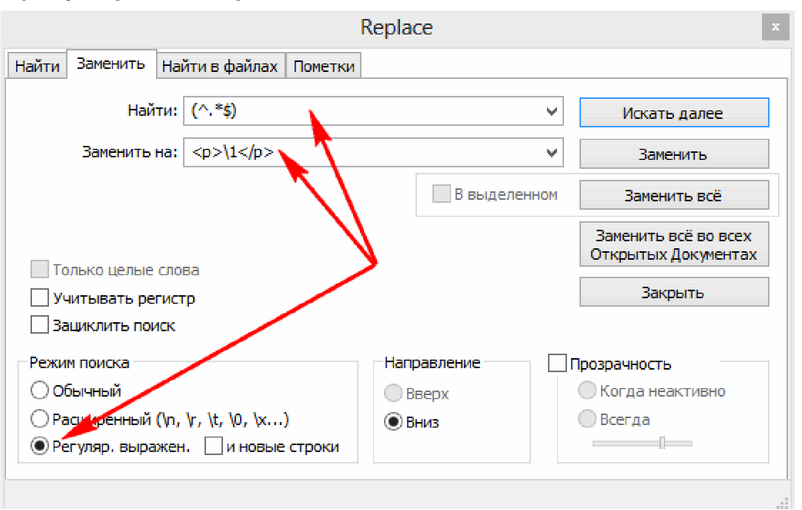


Рисунок 1. Графическое изображение примера поиска поиска и замены абзаца

Важное отступление

Если в шаблоне поиска («Найти») встречаются символы схожие с метасимволами, то их нужно экранировать квадратными скобками. Например, в искомом тексте нужно найти именно вот такое сочетание **123*456**, то есть в нём есть символ *****, тогда в поле «Найти» должно быть **123[*]456**. В противном случае, будет искаться сочетание вида **1233456**, так как *означает повторитель.

В примере «**Поиск фразы только в середине текста**», точка в **в ред. Федерального закона** не экранирована, так как в данном случае не принципиально будет искаться именно точка или любой другой символ.
Если предположить, что в тексте могут быть фразы вроде - **в ред: Федерального закона** и такие фразы нужно не учитывать при поиске, то тогда, точку следует указать явно -
^.* в ред[.] Федерального закона .*$.

Пример_1

```
Находит новую строку и добавляет к ней пустую строку
найти:
(^.*$)
заменить:
\1\n\r
```

```
Находит конец строки и добавляет к ней тег <br>
найти:
$
заменить:
<br>\1
```

```
Находит новую строку и заключает её в теги абзаца <p></p>
найти:
(^.*$)
заменить:
<p>\1</p>
```

```
Находит новую строку и заключает её в теги списка <li></li>
найти:
(^.*$)
заменить:
<li>\1</li>
```

Поиск фразы только в середине текста[3]
Найти в тексте строки в которых может быть любой текст как до начала фразы **в ред. Федерального закона,** так и после этой фразы до конца строки
найти:

```
^.* в ред. Федерального закона .*$
```

[3] Пример может сначала показаться лишенным смысла, так как эти же самые условия можно получить просто указав **в ред. Федерального закона**, однако возможно вам необходимо:

- чтобы поиск не учитывал фразу **в ред. Федерального закона**, если именно с неё начинается строка;
- чтобы поиск не учитывал фразу **в ред. Федерального закона**, если именно ей оканчивается строка;
- вы собираетесь до и после искомой фразы добавить что-то такое эдакое, например начало новой строки, чтобы в дальнейшем заменить её ещё на что-либо.

В этих случаях, простой поиск уже не сработает. Хотя возможно это выражение можно сократить или как то усовершенствовать.

Удалить все предложения, в которых стоит запятая после первого слова

Необходимо удалить все предложения, в которых стоит запятая после первого слова в этом предложении.

Пример_2

1. Во вкладке поиска «**Пометки**» устанавливаем чек-бокс «**помечать закладкой**» (Рисунок 2)
2. В поле найти добавляем:
(^[а-я]*,)
3. После выставления в тексте пометок, выполняем команду:
«Меню/Поиск/Закладки/Удалить все строки С закладкой»

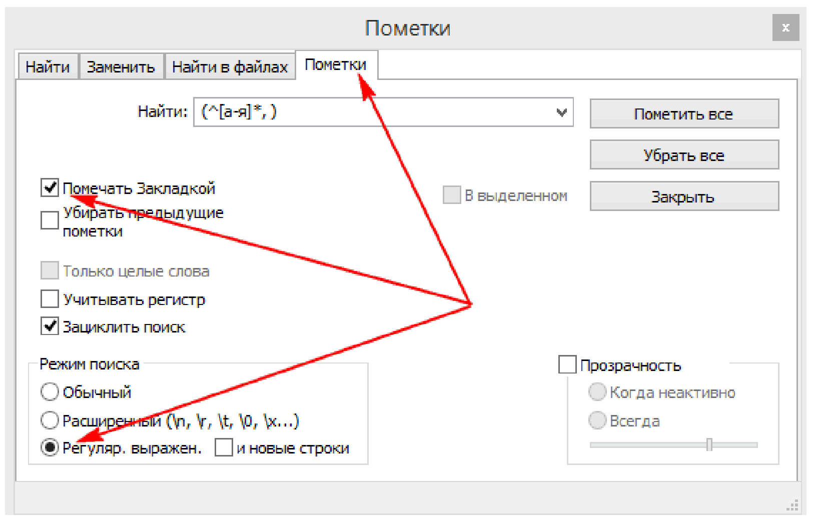


Рисунок 2. Удалить все предложения, в которых стоит запятая после первого слова

Как очистить историю «поиска» в Notepad++

Давно пользуюсь «поиском и заменой», и столько же меня раздражала одна маленькая особенность, с которой я мирился до поры до времени.

А именно, то, что редактор ведет историю «поиска и замены», и при следующем поиске – автоматически предлагает ей (историей) воспользоваться , что безусловно очень удобно в рамках правки одного документа. Все заготовки как говорится «под рукой», не нужно ничего вводить заново.

Однако не всегда это является плюсом, порой после этот список весьма обширен, но уже совершенно не нужен и лишь отвлекает. Раньше старался просто не обращать внимания на эти мелкие неудобства.

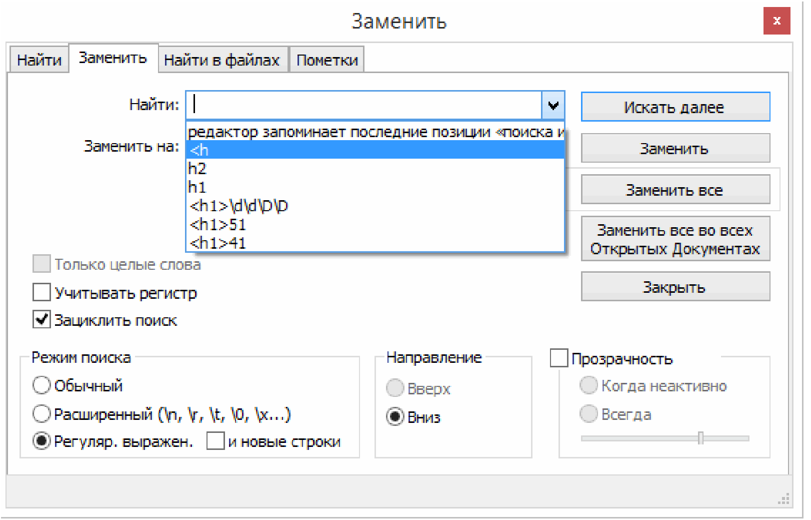


рисунок 3

Недавно узнал как очистить список истории «поиска и замены», не скажу что этот способ очень быстрый и удобный, но может быть кому-то пригодится.

Вернее он кажется долгим, так как нужно идти в скрытый каталог, там что искать. Однако если где-нибудь сохранить путь к файлу, то процесс этот занимает секунд 30 примерно.

> Важно!
> Нижеперечисленные действия нужно выполнять в любом другом редакторе, но не в Notepad++, можно в системном блокноте.

1. Закрываем notepad++;
2. Открываем файл `config.xml` `(%User%\Application Data\Roaming\Notepad++\config.xml)` или как-то так ...

   У меня полный путь выглядит так - `C:\Users\User\AppData\Roaming\Notepad++\config.xml`, здесь все зависит от вашей ОС, а также от способа установки Notepad++, может быть файл где сохраняется история находится непосредственно в директории программы;

3. Удаляем всё что между тегами `<FindHistory></FindHistory>` – рисунок 4.
4. Сохраняем изменения, закрываем блокнот;
5. Открываем notepad++ без истории.

config — Блокнот

Файл Правка Формат Вид Справка

```
<?xml version="1.0" encoding="Windows-1252" ?>
<NotepadPlus>
    <FindHistory nbMaxFindHistoryPath="10" nbMaxFindHistoryFilter="10" nbMaxFindHistoryFind="10" nbMaxFindHistoryReplace="10" match
directionDown="yes" fifRecuisive="yes" fifInHiddenFolder="no" dlgAlwaysVisible="no" fifFilterFollowsDoc="no" fifFolderFollowsDoc="n
transparencyMode="0" transparency="150" dotMatchesNewline="no">
        <Find name="&#x0440;&#x0435;&#x0434;&#x0430;&#x043A;&#x0442;&#x043E;&#x0440;  &#x0437;&#x0430;&#x043F;&#x043E;&#x043C;&#x043
&#x043F;&#x043E;&#x0441;&#x043B;&#x0435;&#x0434;&#x043D;&#x0438;&#x0435;  &#x043F;&#x043E;&#x0437;&#x0438;&#x0446;&#x0438;&#x0438;;
&#x00AB;&#x043F;&#x043E;&#x0438;&#x0441;&#x043A;&#x0430;  &#x0438;  &#x0437;&#x0430;&#x043C;&#x0435;&#x043D;&#x044B;&#x00BB;" />
        <Find name="&lt;h" />
        <Find name="h2" />
        <Find name="h1" />
        <Find name="&lt;h1&gt;\d\d\D\D" />
        <Find name="&lt;h1&gt;51" />
        <Find name="&lt;h1&gt;41" />
        <Replace name="&#x043D;&#x0435;  &#x043D;&#x0443;&#x0436;&#x043D;&#x043E;  &#x043D;&#x0438;&#x0447;&#x0435;&#x0433;&#x043E;
&#x0432;&#x0432;&#x043E;&#x0434;&#x0438;&#x0442;&#x044C;  &#x0437;&#x0430;&#x043D;&#x043E;&#x0432;&#x043E;" />
        <Replace name="h2" />
        <Replace name="&lt;h1&gt;" />
        <Replace name="&lt;h1&gt;([&#x0410;-&#x042F;])" />
        <Replace name="&lt;h1&gt;[&#x0410;-&#x042F;][&#x0430;-&#x044F;]" />
        <Replace name="([&#x0430;-&#x044F;])" />
    </FindHistory>
    <History nbMaxFile="0" inSubMenu="no" customLength="-1" />
```

рисунок 4